EXPERIMENTAL EVALUATION OF INFORMATION RETRIEVAL THROUGH A TELETYPEWRITER

by

Morris Rubinoff*†
Samuel Bergman*
Winifred Franks*
and
Elayne R. Rubinoff*

June 1967

UNIVERSITY of PENNSYLVANIA
The Moore School of Electrical Engineering
PHILADELPHIA, PENNSYLVANIA 19104

University of Pennsylvania
THE MOORE SCHOOL OF ELECTRICAL ENGINEERING
Philadelphia, Pennsylvania

EXPERIMENTAL EVALUATION OF INFORMATION RETRIEVAL THROUGH A TELETYPEWRITER

by

Morris Rubinoff*†
Samuel Bergman*
Winifred Franks*
and
Elayne R. Rubinoff*

June 1967

*The Moore School Information
Systems Laboratory
University of Pennsylvania

†Also with
Pennsylvania Research Associates, Inc.

The work described in this paper has been supported by the Air Force Office of Scientific Research, Information Sciences Directorate, and by the Army Research Office-Durham.

Contract AF 49(638)-1421

University of Pennsylvania
THE MOORE SCHOOL OF ELECTRICAL ENGINEERING
Philadelphia, Pennsylvania

EXPERIMENTAL EVALUATION OF INFORMATION RETRIEVAL THROUGH A TELETYPEWRITER

1. INTRODUCTION

The last decade has seen the growth of widespread computer use in a large variety of applications in many different fields of specialization. As a result, the suppliers of computing services have been plunged more and more deeply into the task of providing small amounts of computing time for large numbers of users. In general, two solutions are possible: provide a small computer for each individual user or local group of users, or a few large computers providing service wholesale to many users at many remote terminals. Compromises between the two are possible, and sometimes even desirable. Examples are large secondary memories shared among many small computers, and remote terminals of a large computer with computing capabilities of their own.

Information systems (i.e., "mechanized libraries") designed to make large files of information easily accessible to many users, most of whom require infrequent access, are forced into the category of large-processor terminal systems. The teletypewriter (TTY) is currently the least expensive and most widely distributed remote terminal. Commercial communication lines provide relatively inexpensive remote communication for infrequently used terminals. However, the TTY-commercial line combination presents a number of problems to the designers of information systems. Some of these are discussed in this paper. Most particular attention is paid to the interactive (man-machine) language for TTY, and some of the mechanical and psychological limitations in their design.

The primary search activities of a library user can be subdivided into three distinct categories: first, he may desire to ask for an explanation of the indexing or classification system used, or for a guide to the index terms or classification subject headings, or for some portion of these which holds his particular interest; second, he may search the catalog (or index) system for accession numbers of documents which are likely to fulfill his needs; and third, he may withdraw documents thus pinpointed from the shelves for browsing or decision as to actual relevance to the problem at hand.

These three search activities may be repeated any number of times in some arbitrary sequence until the search has been completed. Depending on the thoroughness of the search, the searcher reaches a certain level of confidence that he has a list of accession numbers, and possibly a set of documents, which will (best) fulfill his information requirements.

The current research project has engaged in some experiments on information retrieval from a computerized file where communication was established through an on-line typewriter. An information retrieval system which provides for man-machine dialogue at a remote inquiry terminal should provide a searcher with many of the tools which would be available to him were he actually performing this search at a library or repository of documents. He should have the capability of tracking down documents of interest to him in much the same way he would do "in person" and probably with greater ease. To this end, the system should provide him with the same three primary search tools:

(1) Library structure: information about the organization of the document file and access to any indexing or classification tools employed;

(2) Document retrieval: ability to partition the document file into sets of documents of varying degrees of relevance to his search, as a function of the information provided by the searcher; and

(3) Document perusal: ability to "peruse" documents which might be of interest, or at least to request certain information about such documents, such as bibliographic data or abstracts.

A mechanized library should provide a remote user information about its structure as well as instruction in its use. Most important, the scheme employed in retrieval of documents, or in partitioning the document file into document sets of varying relevance with respect to a particular request, should be readily available to the user. If a classification scheme is employed, the "classification tree" should be available for inspection by the user upon request and in an informative and easily digestible form. In a coordinate indexing environment with controlled vocabulary, the user should be provided, if possible, with means to inspect the thesaurus, together with any semantic tools used in indexing. Furthermore, instruction in the use of the system and its command language should be available in some form. If possible, the system should "bootstrap the user in," or assist a neophyte user who has no previous experience, much the same as the librarian instructs a new user to the traditional library.

These characteristics become not only necessary, but vital, in an open system, where any teletypewriter with data-phone connection may be used as a remote terminal. At some time in the future, such an information system and its command language may be widespread, and its rules of operation will be common knowledge. Until such time, the system had better be simplicity itself if it is in any way to compete with standard library research technique [1,5,7,9,13].

Clarity is important because results should be understood the first time they are printed. Printout takes long enough as is, and although concise, the format should not leave room for misunderstanding. Codes or abbreviations used should be carefully defined, and avoided where ever possible. The balance between clarity and conciseness is not always apparent.

2. DOCUMENT PERUSAL

The traditional library provides the user with the ability to peruse documents located (e.g., removed from the shelf) as a result of a search. Further, card catalogue cards may be perused for subject index and bibliographic data and sometimes short descriptions. Reviews, summaries, and abstracts may be perused in appropriate journals. These procedures assure that material checked out or ordered from the library is relevant and of interest. A remote terminal system must provide similar facilities. Although perusal of documents themselves is somewhat beyond the scope of the present state of the art, compact descriptive material for each document must be available for perusal. This information must necessarily require only small amounts of storage over and above that already required for retrieval. A duality exists here--- information characterizing a document for retrieval characterizes it for perusal as well. Bibliographic data, index terms, and abstracts are all useful in both retrieval and perusal. The difficulty lies in file structure design and formatting so that the desired information can be requested in a reasonable way and printed in a reasonable format within reasonable time limitations.

A remote information system operating in conversational mode with document perusal available in real time allows the user to "zero in" on the documents of interest to him, and to choose with a high degree of confidence. The actual retrieval algorithms or procedures are of less importance, since the retrieval procedure is interactive and iterative.

3. COMMAND LANGUAGE

A mechanized information retrieval system must employ a command language. Such a language can be classified according to its orientation, i.e., system-oriented language would presumably employ many curt, arbitrary symbols to represent various operations, and a syntax closely akin to the logical structure of the mechanized system. A language which is more user-oriented would be provided with syntax or construction of statements and requests more akin to the logical formulation of requests by people. Certainly, the most user-oriented language possible is natural English, or any other natural language that a searcher would ordinarily employ for communication with a librarian. Of course, there are varying degrees of user and system orientation.

Command languages can also be measured according to their flexibility or general applicability. A large variety of test or pilot information systems employ different specialized capabilities. A system that can be thought of as having some sort of general applicability should easily accommodate differing points of view or system capabilities, such as binary or multiple relevance rating capabilities, and a large set of retrieval specification parameters (bibliographic specification, key word in context, subject index, coordinate index, etc.).

Naturally, there is a tradeoff of time, load capability and/or efficiency with respect to the d[illegible] of flexibility and user orientation of a command language. A language which allows statements in a limited subset of English is necessarily slow and requires large scale mechanization and a large storage just for passing and translating statements. A highly efficient, limited capability language requires greater learning time on part of the user, and therefore might risk a smaller subscribership. Some balance is necessary in the choice of a language commensurate with its intended use and yet consistent with available computation and storage devices. This is a particularly important question in a system designed for thousands of users, many of whom need access very infrequently (maybe only once or twice annually). A language that is difficult to learn would drastically repel users who require only occasional access.

4. THE MOORE SCHOOL EXPERIMENTAL INFORMATION SYSTEM

An information system responsive to many of the preceding guidelines has been set up by the Moore School Information System Laboratory (MSISL), University of Pennsylvania. The document set consists of the Repository of the Association of Computing Machinery (about 1500 documents). The interactive language includes a section of computer-directed search with computer control exercised through a sequence of "yes-no" type questions and a message insertion section with a special request language ("Symbolic Language") employed by the searcher in specifying a document retrieval request. Three types of requests are available, termed "Retrieve", "Combine", and "Display". "Retrieve" allows the searcher to specify a retrieval request by a logical combination of descriptors; "Combine" elicits retrieval of a set of documents characterized by one or more of a set of descriptors; and "Display" causes retrieval of a set of documents specified by accession number.

It is sufficient for present purposes to refer to Fig. 1 for a syntactic description of Symbolic Language, Figs. 2a-2d for an illustration of an actual on-line search, and Fig. 3 for a semantic description of various elements of a retrieval request. In Figs. 2a and 2b a request is made for documents characterized by one or more of the five items IBM, PROGRAMMING, MANUAL, FORTRAN, and FEB 1955. The first four are supposedly index terms and the fifth is a

Figure 1

Backus Normal Specification of Symbolic Language Syntax

<REQUEST>::=<RETRIEVE REQ> <END> | <COMBINE REQ> <END> | <DISPLAY REQ> <END>

<RETRIEVE REQ>::= RETRIEVE <R SPECS>

<R SPECS>::= <CATEGORY> <TERM> | <R SPECS> <OPERATOR> <R SPECS>

<OPERATOR>::= +|&|'

<CATEGORY>::= $A0|$A1|$A2|$A3|$A4|$A5|$A6|$A7|$A8|$A9|$B|$C

<TERM>::= <STRING> | (<TERM> <OPERATOR> <TERM>)

<STRING>::= <WORD> | <STRING> <WORD>

<WORD>::= Any set of up to 24 contiguous characters from the alphabet, the decimal digits, hyphen, asterisk, period, comma, apostrophe.

<COMBINE REQ>::= COMBINE <C SPECS>

<C SPECS>::= <CATEGORY> <C SEQUENCE> | <C SPECS> / <C SPECS>

<C SEQUENCE>::= <STRING> | <C SEQUENCE> / <STRING>

<DISPLAY REQ>::= DISPLAY (<ACCESSION LIST>)

<ACCESSION LIST>::= <ACC NUMBER> | <ACCESSION LIST>, <ACC NUMBER>

<ACC NUMBER>::= <NUMBER> | <NUMBER> - <DIGIT>

<DIGIT>::= One of the ten decimal digits.

<NUMBER>::= An integer, from one to four digits.

<END>::= The sequence of two teletypewriter symbols "< >".

Note: The elements of a Symbolic Language statement as specified above should be separated by one or more blanks. Carriage returns are considered blanks, so that elements may not be continued from one line to another. The one exception to this rule is in writing accession numbers with appended digits. In this case, there are no blanks inserted, as for example -- 953-7. Furthermore, a character "←" immediately following it; a line of characters may be erased by typing "←!"; an entire message may be erased by typing "←?".

TERM / SYNONYMS	Assembler	Assembly Routine	Command	Computer Code	Computer-Dependent Language	Function	Generator	Instruction Code	Interpreter	Interpretive Routine	Machine Instruction	Machine Language	Machine Operation	Machine-Oriented Language	Operation Part	Order	Order Code	Procedure	Pseudo-Instruction	Routine
Assembly Program	X	X																		
Computer Instruction			X								X					X				
Computer Instruction Code				X				X				X					X			
Computer Language					X							X		X						
Computer Operation						X							X							
Computer-Oriented Language					X							X		X						
Directive			X								X					X			X	
Generating Program							X													
Instruction			X								X					X				
Instruction Set								X									X			
Interpretive Program									X	X										
Operation Code				X											X		X			
Program																		X		X

Condensed Sample of Classification Table for Synonyms

(from table approximately six times this size)

Figure 2a

TERM \ WORDS IN DEFINITION	Code	Computer	Construction	Elementary	Instruction	Language	Operation	Program	Programming	Specification	Symbol	Translation
Assembly Program	X					X	X	X			X	X
Computer Language		X			X	X	X		X			
Generating Program		X	X					X			X	
Instruction		X		X						X	X	
Operation Code	X	X		X			X			X	X	

TERM-WORDS IN DEFINITION

GENERIC \ SPECIFIC	Assembly Language	Assembly Program	Auto Code	Computer Instruction	Directive	Instruction Code	Interpretive Program	Jump Command	Operation Code	Problem-Oriented Language	Procedure-Oriented Language	Programming Language	Pseduo Code	Translating Program
Code			X			X			X				X	
Command								X						
Instruction				X	X									
Language										X	X	X		
Program		X					X							X
Programming Language	X		X										X	

GENERIC-SPECIFIC

Condensed Samples of Classification Tables

Figure 2b

WHOLE \ PART	Assembly Program	Computer Instruction	Diagnostic Program	Generating Program	Instruction	Operation	Picture	Pragmatics	Semantics	Statement	Symbol	Syntax
Compiling Program	X		X	X								
Computer Instruction						X						
Computer Language		X										
Generating Program	X											
Language					X			X	X	X	X	X
Programming Language					X		X			X		

WHOLE-PART

NOUN \ MODIFIERS	Absolute	Assembly	Automatic	Complete	Computer	Diagnostic	Instruction	Interpretive	Machine	Macro	Operation	Order	Problem-oriented	Programming	Translating
Code					X		X		X		X	X			
Instruction	X			X	X				X	X					
Language		X			X				X				X	X	
Operation			X		X				X						
Program		X			X	X		X		X					X

NOUN-MODIFIERS

Condensed Samples of Classification Tables

Figure 2c

MEANS \ END	Construction	Control	Description	Execution	Interpretation	Preparation	Production	Representation	Specification	Translating	Unconditional transfer
Assembly Program					X		X			X	
Computer Instruction				X					X		
Directive		X									
Interpretive Program				X						X	
Jump Command											X
Operation Code								X	X		

MEANS-END

Condensed Sample of Classification Table

Figure 2d

Figure 3

Semantic description of Symbolic Language

A string (<STRING>) is a string of words comprising the descriptor of one or more documents. A substring of such a descriptor is also acceptable as a descriptor, e.g., part of a title or one word of a two word term.

Accession numbers (<ACC NUMBER>) are unique numbers in the format defined in Fig. 1 assigned to each document in the collection.

The words "RETRIEVE", "COMBINE", and "DISPLAY" indicate the type of retrieval desired.

The following action codes indicate categories under which documents are indexed (<CATEGORY>):

$A0 - Date of indexing and indexer's code
$A1 - Authors
$A2 - Date of Issue
$A3 - Title
$A4 - Editors, etc.
$B - Index terms
$C - Added information (Special codes provide a large variety of additional information about documents. Code list is available).

$A5 - Issuers
$A6 - Page size
$A7 - Number of illustrations
$A8 - Number of pages
$A9 - Collection in which article appears

The following infix binary connectives are used to combine terms (<Operator>)

&	AND	(logical <u>and</u>, &)
+	OR	(logical exclusive <u>or</u> V)
↑	AND NOT	(logical <u>co-implication</u>, or <u>inhibition</u>, $\not\supset$)

A list of accession numbers of documents is actually produced as a result of a retrieval request. The searcher is then queried as to what (if anythin) he would like to peruse (have printed out) concerning these documents on the list publication date. After the retrieval, the accession numbers of those characterized by at least three of these five items is requested and printed. Figure 2c illustrates the retrieval of four documents authored by either PATTERSON or CARR, J.W. and the printout of all bibliographic information available concerning these documents. Figure 2d the authors (where available) and titles of four documents listed by accession number are printed.

Some of the important features of the system are as follows:

1. Response time of the system to most requests (unless they are exceedingly complex, and unlikely in a real library search) is on the order of one to five seconds.
2. Messages may be edited while typing (see note on Fig. 1). They may also be edited after the message is completed, if desired.
3. The descriptors used as characterization of a document for retrieval purposes are available for perusal of retrieved documents.
4. Retrieval and perusal may be alternated any number of times in deciding upon exactly the desired documents.

The experimental system is not time-shared, but is accessible to any teletypewriter with a data-phone connection. Tests have been run from as far as Boulder, Colorado, by standard telecommunication lines. (The system is of course located in Philadelphia at the University of Pennsylvania.)

5. SOME QUESTIONS

Some subjective as well as objective questions arose with regard to the MSISL system. In such a system easy to use? Does a searcher feel comfortable using it? How quickly can a person with no background in information retrieval learn to use the system? What form of instruction in its use is most fruitful? How effective is the retrieval language for efficient and relevant retrieval?

Where does it fail to emulate the traditional library search?

Noting that such answers with respect to a small pilot system do not necessarily provide the answers with respect to a large multi-user system with a large document collection, but might nonetheless provide some clues as to larger system design, several experiments were performed. These experiments did in fact provide some guidelines for future design of interactive systems of this sort.

6. THE EXPERIMENTS

The experiments were conducted:

I. The purpose of this test was to study the ease of familiarization with the capability of the system, the ease of developing working retrieval strategy, and to reveal the drawbacks of the system from the psychological viewpoint of the user.

Subjects were given a retrieval task, told of the capability of the retrieval system, but not taught the command language. They verbally transmitted requests to a teletype operator who translated these to the command language. The subjects saw the system output in this way formulated further requests. Their gradual familiarization with the system was then studied by an examination of the development of their search strategy.

Appendix A presents the instructions given to the subjects for the purposes of this test.

II. The purpose of this test was to study the ease or difficulty of teaching the request (command) language. Appendix B presents the instruction sheet given subjects in this test.

Subjects were given a written booklet of instructions for use of the system, including the syntax and semantics for the command language. The programmed instruction routine was still available at the typewriter keyboard.

III. The purpose of this test was similar to that of test II, but was also intended to reveal whether complete instructions for the use of the system could be assimilated or partially assimilated before actual trial.

Subjects were given a retrieval task and told nothing about the system except for the initial identification code and a general description of its capabilities. A programmed instruction routine was included in the system which, sensing no acceptable input message after initial identification, inquired as to whether the operator desired instruction in the use of the system. An affirmative reply elicited the teaching routine.

In addition to a study of the actual interactive dialogue, each of the subjects was interviewed subsequent to the experiment.

7. CONCLUSIONS AND FUTURE PLANS

The combined results of the three experiments can be summarized as follows:

a) A minimum of 30 minutes is required to learn even the rudiments of the command language (Symbolic Language). This did not vary substantially between graduate students in the sciences and securities, between those familiar with programming languages and those not. Often, a good deal of time is necessary for familiarization with the teletype keyboard.

b) As in most other tasks, although written instructions are helpful, practice is essential.

c) The programmed instruction routines, which took approximately 20 minutes, were mostly of the "yes-no" variety. Expanded routines, which would require other replies and would give student-users actual practice in formation of retrieval requests would have been better, but would have taken considerably more time (and would have tried the patience of more astute students).

d) Flexibility was sacrificed for the sake of simplicity in the command language, and this was pointed out as a definite deficiency by many of the subjects. Increased flexibility would have, of course, added to the required learning time.

e) The slow typing speed of the teletypewriter (slower than the average reading rate) and the high noise level were pointed out as distracting factors by many of the subjects.

Some of the difficulties encountered can certainly be eliminated by redesign. Careful choice of messages can assure both conciseness and clarity. Back-up messages can explain in greater detail what the concise messages leave unclear to any user. Provisions may be made for termination of long printout in "Midstream" when the user discovers he is on the wrong track or in error.

Certain difficulties, however, are inherent in the limitations of the teletypewriter as a remote console. Reading rates are much greater than the printout rates, particularly in long printouts. Often the contents of much of the printout is guessed before it actually appears on the paper. For instance, when inquiring for the publisher of a book, the letters "MCGR" often suffice to convey "MCGRAW HILL BOOK COMPANY", and the remaining two seconds of printout are not only wasted, but are annoying to the user. Furthermore, the noise level of the teletypewriter is high and promotes anything but clear thinking and concentration. Backspacing and erasures are not possible, and the "←" editing feature does not entirely make up for this deficiency. These are problems we intend currently to live with, hoping that the advantages of remote information consoles offset the disadvantages.

The command language is more readily adaptable. It is planned to replace Symbolic Language, employed in this experiment, by a command language called "Real English," which employs real English sentences in a specified format. Pilot tests show that such a request language is powerful enough to handle a majority of requests, and is easily mastered by neophytes.

Without entering into the syntactic-semantic structure of the "Real English" developed to date, we shall present a few examples of a request statement which are acceptable statements in the language. All five statements below will cause the same retrieval, and are considered identical by the system after passing.

> PLEASE LOCATE EVERYTHING WRITTEN BY ROBERT PERKINS ABOUT EASIAC OR PSEUDO-COMPUTERS BETWEEN 1955 and 1859 < >
>
> COULD YOU FIND FOR ME SOMETHING CONTAINED IN THE REPOSITORY CONCERNING EASIAC OR PSEUDO-COMPUTERS THAT WAS AUTHORED BY ROBERT PERKINS AFTER 1954 AND BEFORE 1960 < >
>
> I NEED ALL THE AVAILABLE DOCUMENTS PUBLISHED DURING THE PERIOD 1955 TO 1959 BY ROBERT PERKINS ON THE SUBJECTS OF EASIAC OR PSEUDO-COMPUTERS < >
>
> WE'RE INTERESTED IN HAVING REFERENCES AND MATERIAL ON EITHER PSEUDO-COMPUTERS OR EASIAC AUTHORED BY ROBERT PERKINS FROM 1955 TO 1959 < >
>
> I WOULD LIKE YOU TO HELP ME OBTAIN INFORMATION FROM YOUR LIBRARY RELATED TO EASIAC OR PSEUDO-COMPUTERS AND WRITTEN BY ROBERT PERKINS IN THE YEARS 1955 THROUGH 1959 < >

Instruction routines in the use of this command language will be built into the system and further tests will be run to check their effectiveness. Unacceptable statements will cause the system to query the user as to his intentions, ask for synonyms of words not included in the system's semantic dictionary, and request the user to restate his question in different words. Hopefully, this will not occur often.

APPENDIX A

Directions for Test Study I

THE PURPOSE: We are testing the effectiveness of the present information-retrieval apparatus. You are helping us test the system performance in a simulated retrieval situation.

THE TASK: We would like you to compile a list of documents which will be helpful in writing a paper on "Artificial Languages." The paper will discuss numerous languages and will mention such items as algebraic languages, list processing, and business-oriented languages. The paper will emphasize the history of the development of automatic programming. You should use the retrieval system to collect the information.

THE METHOD: The following explanation describes the system, and it indicates what material can be obtained and what request forms you may use to obtain it.

In the ACM Repository there are approximately 1500 documents of many types: journals, articles, manuals, mimeographed reports, and others, on the subject of computing machinery. Each of these documents, distinguishable from every other by even so little as a different color of cover, has been given an identifying accession number. Every document has been described as completely as possible by its bibliographic elements (Group A), the topics with which it deals (Group B), and the values assinged to certain of its attributes (Group C). Some or all of this descriptive material is available for every document which has been entered in the system and which, therefore, has an accession number. The information categories describing a document are as follows:

Group A - A1 Authors
A2 Date of publication
A3 Title
A4 Editors or contributors
A5 Publisher or issuing agency
A6 Physical dimensions in cm.
A7 Number of pages
A8 Number of illustrations
Group B ---- Index terms
Group C ---- Added Information (See Partial Key to Code Used for "Added Info" for the ACM Repository)

The information retrieval system that you will be using will refer you to those documents which appear to be related in some way to your problem. If, for example, you request information about "programming," the system will retrieve all documents which have been indexed under that term. Similarly, you may ask for all articles by a certain author, with certain words in their titles, from a particular source, produced during a certain period, etc.

At the present time the system will retrieve material in answer to three forms of logical requests: "and," "and not," "or". That is, you may ask for all documents written by John Doe and on disk files (conjunction); you may ask for all documents written by John Doe and not about disk files (negation); or you may ask for all documents written by John Doe or about disk files (disjunction). You may also use combinations of these logical expressions. An example might be: documents (by John Doe or George Smith) and (on peripheral storage and not disk files) and (written in 1960 or 1961 or 1962) and not (published by IBM). After a retrieval has been executed, you will be informed how many documents have been "retrieved." You may then request the print-out of any or all of the information categories describing these documents.

In addition to these logical request forms, you may use a "combinatorial" request, involving up to 8 terms. For example, Combine (author) John Doe, (index term) disk files, (index term) peripheral storage, and (publisher) IBM.

In this case, the system will inform you how many documents indexed by 4 of the terms have been retrieved, how many indexed by 3, by 2, and by 1. You will then be given the opportunity of seeing a print-out of information on documents included in any of the 4 sets. Please note: logical operators (and, and not, or) may not be incorporated in a combinatorial request.

It should be noted that since the quality of indexing has not always been what it should be and since the system does not yet recognize synonymic expressions, you will be well advised to try different methods of requesting information whenever the results of a search are not pleasing to you. If you can think of more than one term for the subject under search (programming, coding, language), be sure to use all possibilities.

As we have stated earlier, suppose, for the purposes of this experiment, that you are writing a paper on "Artificial Languages," including algebraic languages, business-oriented languages, list processing, etc. The emphasis of your search is to be on the history of the development of automatic programming.

The goal of your search is a list of accession numbers for documents which you think will be of value in writing this paper. Attempt to make this list as complete as possible; at the same time, on the basis of what you can determine from the document descriptions, try to make your final list as precise as you can (that is, as pertinent to the problem as possible).

You will be allowed 20 minutes in which to communicate with the system through the teletypewriter operated by Shelly Newman. You should work as quickly as possible during this time, but you should not necessarily consider speed as your aim. The completeness and precision of your final list is most important. Although it is expected that your search techniques will improve as you become more familiar with the system, you should realize that this experiment is being performed as a test of the system and not of your research abilities.

After you have completed the 20 minutes of communication with the system, you will receive the print-out of it from the teletypewriter and will be allowed 10 minutes in which to review the print-out and decide on your final list of accession numbers.

Your participation in this experiment is greatly appreciated. Thank you.

APPENDIX B

DIRECTIONS FOR TEST STUDY II

PURPOSE: We are conducting studies of a computerized information retrieval system. The retrieval system consists of a computer and a teletypewriter. Stored in the computer is descriptive information on 1500 documents, including journals, articles, manuals, reports, and lecture notes, on the subject of computing machinery and programming. Using the teletypewriter as a means of communication, a person can type a request for different kinds of information regarding the various documents. After processing the request, the computer will retrieve the information from its files and then print the information with the teletypewriter. If desired, the user can continue to make requests and obtain information. In this way, a person can quickly and comfortably retrieve information about library documents. The computer does the actual searching and sorting.

We would like you to help us test the capabilities and accuracy of computer performance by requesting information from the system.

THE TASK: Assume that you are writing an essay on the history and development of FORTRAN, a computer programming language. You are looking for several documents which will be helpful research material for the essay. You learn that the retrieval teletypewriter will be unemployed for half an hour or so. Being eager to start research, you can now use the available teletypewriter to find titles and other information about documents describing FORTRAN. This retrieved information can lead you directly to valuable research material shelved in the engineering library.

In order to be using the teletypewriter-computer communication, you type the following:

1 -- In answer to I AM:=, type your identification number as follows:

003333 < >

2 -- In answer to THE OPERATING MODE IS:=, type the following:

SEARCH < >

3 -- In answer to YOU MAY PROCEED:=, begin the search.

Note that you must type < > to indicate the end of your messages. Otherwise, the computer will not recognize that you have finished typing..

Furthermore, when you desire additional guidance, the computer will provide assistance and explanations.

After you have found titles of three documents that you think will be helpful for an essay on FORTRAN, please record the titles on a piece of paper, and present it to the experimenter. We are interested in seeing if the computer retrieves relevant information and prints information accurately.

Your participation in this test study is greatly appreciated. Thank you.

REFERENCES

1. Rubinoff, M. and White, J. F., Jr.; Establishment of the ACM Repository and the Principles of the IR System Applied to its Operation. Comm. ACM, 8:595, 1965.

2. Rubinoff, M. and White, J. F., Jr.; Description of Cataloging and Indexing System for the ACM Repository. The Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia. 1965.

3. Rubinoff, M.; A Look Ahead. Proceedings of the Second National Symposium on Engineering Eduction. Engineers Joint Council, October 1965.

4. Rubinoff, M., et al; The Moore School Information Systems Laboratory. The Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia. June 1966.

5. Fischer, Stephen B.; An Executive Control System for Information Retrieval via a Remote Console. M. S. Thesis presented to The Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia. 1966.

6. Rubinoff, Elayne R.; Factors Involved in Testing Information Systems. The Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia. August 1966.

7. Lowe, Thomas C.; Design Principles for an On-Line Information Retrieval System. Ph.D. Dissertation presented to The Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia. 1966.

8. Rubinoff, M.,et al; Summary Description of Easy English. The Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia. February 1967.

9. Simmons, R. E., Answering English Questions by Computer: A Survey. Comm. ACM 8, No. 1, 1965.